Monitoring Event Logs

eG Enterprise v6.0

Restricted Rights Legend

The information contained in this document is confidential and subject to change without notice. No part of this document may be reproduced or disclosed to others without the prior permission of eG Innovations Inc. eG Innovations Inc. makes no warranty of any kind with regard to the software and documentation, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Trademarks

Microsoft Windows, Windows NT, Windows 2003, and Windows 2000 are either registered trademarks or trademarks of Microsoft Corporation in United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Copyright

Table of Contents

Table of Figures

Chapter 1

Monitoring Event Logs

Many applications record errors and events in various proprietary error logs. These proprietary error logs have different formats and display different user interfaces. Moreover, the system administrator cannot merge the data to provide a complete report. Therefore, the administrator needs to check a variety of sources to diagnose problems.

Event logging in Microsoft® Windows NT®/Windows® 2000 provides a standard, centralized way for applications and the operating system to record important software and hardware events. The event-logging service stores events from various sources in a single collection called an *event log*. The system administrator can use the event log to help determine what conditions caused the error and the context in which it occurred. By periodically viewing the event log, the system administrator may be able to identify problems (such as a failing hard drive) before they cause damage.

In order to enable monitoring of the event logs pertaining to a server, the eG Enterprise suite provides for a special server type named *Event Log*. The layer model of an *Event Log* server is as depicted below:

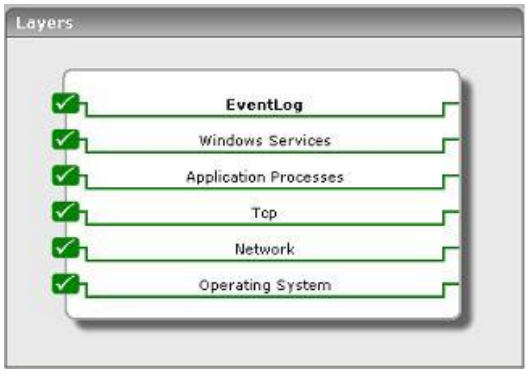

Figure 1.1: The layer model of an EventLog server

The 5 layers at the bottom of Figure 1.1 have been dealt with extensively in the *Monitoring Unix and Windows Servers* document. The following section will throw light on the **EventLog** layer, which is the upper most layer.

1.1 The EventLog Layer

This layer monitors the system, application, and security logs on the Windows host, and reports the number of errors/warnings/general information events that have occurred on the host.

Figure 1.2: Test executing on the EventLog layer

1.1.1 Application Event Log Test

This test reports the statistical information about the application events generated by the target system.

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT –** Refers to the port used by the EventLog Service. Here it is null. 4. **LOGTYPE** – Refers to the type of event logs to be monitored. The default value is *application*. 5. **POLICY BASED FILTER** - Using this page, administrators can configure the event sources, event IDs, and event descriptions to be monitored by this test. In order to enable administrators to easily and accurately provide this specification, this page provides the following options: ➢ Manually specify the event sources, IDs, and descriptions in the **FILTER** text area, or, ➢ Select a specification from the predefined filter policies listed in the **FILTER** box For explicit, manual specification of the filter conditions, select the **NO** option against the **POLICY BASED FILTER** field. This is the default selection. To choose from the list of pre-configured filter policies, or to create a new filter policy and then associate the same with the test, select the **YES** option against the **POLICY BASED FILTER** field. 6. **FILTER** - If the **POLICY BASED FILTER** flag is set to **NO**, then a **FILTER** text area will appear, wherein you will have to specify the event sources, event IDs, and event descriptions to be monitored. This specification should be of the following format: *{Displayname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}*. For example, assume that the **FILTER** text area takes the value, *OS_events:all:Browse,Print:all:none:all:none*. Here: ➢ *OS_events* is the display name that will appear as a descriptor of the test in the monitor UI; ➢ *all* indicates that all the event sources need to be considered while monitoring. To monitor specific event sources, provide the source names as a comma-separated list. To ensure that none of the event sources are monitored, specify *none*. ➢ Next, to ensure that specific event sources are excluded from monitoring, provide a comma-separated list of source names. Accordingly, in our example, *Browse* and *Print* have been excluded from monitoring. Alternatively, you can use *all* to indicate that all the event sources have to be excluded from monitoring, or *none* to denote that none of the event sources need be excluded. ➢ In the same manner, you can provide a comma-separated list of event IDs that require monitoring. The *all* in our example represents that all the event IDs need to be considered while monitoring.

- Similarly, the *none* (following *all* in our example) is indicative of the fact that none of the event IDs need to be excluded from monitoring. On the other hand, if you want to instruct the eG Enterprise system to ignore a few event IDs during monitoring, then provide the IDs as a comma-separated list. Likewise, specifying *all* makes sure that all the event IDs are excluded from monitoring.
- The *all* which follows implies that all events, regardless of description, need to be included for monitoring. To exclude all events, use *none*. On the other hand, if you provide a comma-separated list of event descriptions, then the events with the specified descriptions will alone be monitored. Event descriptions can be of any of the following forms - *desc**, or *desc*, or **desc**,or *desc**, or *desc1***desc2*, etc. *desc* here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.
- In the same way, you can also provide a comma-separated list of event descriptions to be excluded from monitoring. Here again, the specification can be of any of the following forms: *desc**, or *desc*, or **desc**,or *desc**, or *desc1***desc2*, etc. *desc* here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters. In our example however, none is specified, indicating that no event descriptions are to be excluded from monitoring. If you use *all* instead, it would mean that all event descriptions are to be excluded from monitoring.

By default, the **FILTER** parameter contains the value*: all:all:none:all:none:all:none*. Multiple filters are to be separated by semi-colons (;).

Note:

The event sources and event IDs specified here should be exactly the same as that which appears in the Event Viewer window.

On the other hand, if the **POLICY BASED FILTER** flag is set to **YES**, then a **FILTER** list box will appear, displaying the filter policies that pre-exist in the eG Enterprise system. A filter policy typically comprises of a specific set of event sources, event IDs, and event descriptions to be monitored. This specification is built into the policy in the following format:

{Policyname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}

To monitor a specific combination of event sources, event IDs, and event descriptions, you can choose the corresponding filter policy from the **FILTER** list box. Multiple filter policies can be so selected. Alternatively, you can modify any of the existing policies to suit your needs, or create a new filter policy. To facilitate this, a **Click here** link appears just above the test configuration section, once the **YES** option is chosen against **POLICY BASED FILTER**. Clicking on the **Click here** link leads you to a page where you can modify the existing policies or create a new one (refer to page 9). The changed policy or the new policy can then be associated with the test by selecting the policy name from the **FILTER** list box in this page.

	7. **USEWMI** - The eG agent can either use WMI to extract event log statistics or directly parse the event logs using event log APIs. If the **USEWMI** flag is **YES**, then WMI is used. If not, the event log APIs are used. This option is provided because on some Windows NT/2000 systems (especially ones with service pack 3 or lower), the use of WMI access to event logs can cause the CPU usage of the WinMgmt process to shoot up. On such systems, set the **USEWMI** parameter value to **NO**. **On the other hand, when monitoring systems that are operating on any other flavor of Windows (say, Windows 2003/XP/2008/7/Vista/12), the USEWMI flag should always be set to 'Yes'.** 8. **STATELESS ALERTS** - Typically, the eG manager generates email alerts only when the state of a specific measurement changes. A state change typically occurs only when the threshold of a measure is violated a configured number of times within a specified time window. While this ensured that the eG manager raised alarms only when the problem was severe enough, in some cases, it may cause one/more problems to go unnoticed, just because they did not result in a state change. For example, take the case of the EventLog test. When this test captures an error event for the very first time, the eG manager will send out a **CRITICAL** email alert with the details of the error event to configured recipients. Now, the next time the test runs, if a different error event is captured, the eG manager will keep the state of the measure as **CRITICAL**, but will not send out the details of this error event to the user; thus, the second issue will remain hidden from the user. To make sure that administrators do not miss/overlook critical issues, the eG Enterprise monitoring solution provides the **stateless alerting** capability. To enable this capability for this test, set the **STATELESS ALERTS** flag to **Yes**. This will ensure that email alerts are generated for this test, regardless of whether or not the state of the measures reported by this test changes. 9. **EVENTS DURING RESTART** - By default, the **EVENTS DURING RESTART** flag is set to **Yes**. This ensures that whenever the agent is stopped and later started, the events that might have occurred during the period of non-availability of the agent are included in the number of events reported by the agent. Setting the flag to **No** ensures that the agent, when restarted, ignores the events that occurred during the time it was not available. 10. **DDFORINFORMATION** – eG Enterprise also provides you with options to restrict the amount of storage required for event log tests. Towards this end, the **DDFORINFORMATION** and **DDFORWARNING** flags have been made available in this page. By default, both these flags are set to **Yes**, indicating that by default, the test generates detailed diagnostic measures for information events and warning events. If you do not want the test to generate and store detailed measures for information events, set the **DDFORINFORMATION** flag to **No**. 11. **DDFORWARNING** – To ensure that the test does not generate and store detailed measures for warning events, set the **DDFORWARNING** flag to **No**.

<table>
<tr><td></td><td colspan="3">12. DD FREQUENCY - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is 1:1. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying none against DDFREQ.
13. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option. To disable the capability, click on the Off option. The option to selectively enabled/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
• The eG manager license should allow the detailed diagnosis capability
• Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the FILTER configured</td></tr>
<tr><td rowspan="4">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Application errors:
This refers to the number of application error events that were generated.</td><td>Number</td><td>A very low value (zero) indicates that the system is in a healthy state and all applications are running smoothly without any potential problems.
An increasing trend or high value indicates the existence of problems like loss of functionality or data in one or more applications.
Please check the Application Logs in the Event Log Viewer for more details.</td></tr>
<tr><td>Application information count:
This refers to the number of application information events generated when the test was last executed.</td><td>Number</td><td>A change in the value of this measure may indicate infrequent but successful operations performed by one or more applications.
Please check the Application Logs in the Event Log Viewer for more details.</td></tr>
<tr><td>Application warnings:
This refers to the number of warnings that were generated when the test was last executed.</td><td>Number</td><td>A high value of this measure indicates application problems that may not have an immediate impact, but may cause future problems in one or more applications.
Please check the Application Logs in the Event Log Viewer for more details.</td></tr>
</table>

	Application critical errors: Indicates the number of critical events that were generated when the test was last executed.	Number	A critical event is one that an application or a component cannot automatically recover from. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. A very low value (zero) indicates that the system is in a healthy state and all applications are running smoothly without any potential problems. An increasing trend or high value indicates the existence of fatal/irrepairable problems in one or more applications. The detailed diagnosis of this measure describes all the critical application events that were generated during the last measurement period. Please check the Application Logs in the Event Log Viewer for more details.
	Application verbose: Indicates the number of verbose events that were generated when the test was last executed.	Number	Verbose logging provides more details in the log entry, which will enable you to troubleshoot issues better. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. The detailed diagnosis of this measure describes all the verbose events that were generated during the last measurement period. Please check the Application Logs in the Event Log Viewer for more details.

The detailed diagnosis of the *Application warnings* measure, if enabled, describes all the application warnings that were generated during the last measurement period.

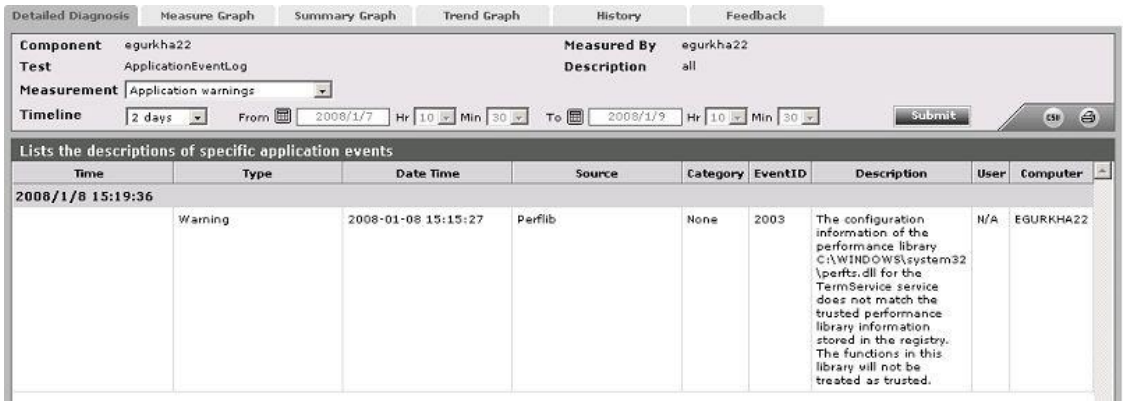

Figure 1.3: The detailed diagnosis of the Application warnings measure

The detailed diagnosis of the *Application information count* measure, if enabled, describes all the general information events that were generated during the last measurement period.

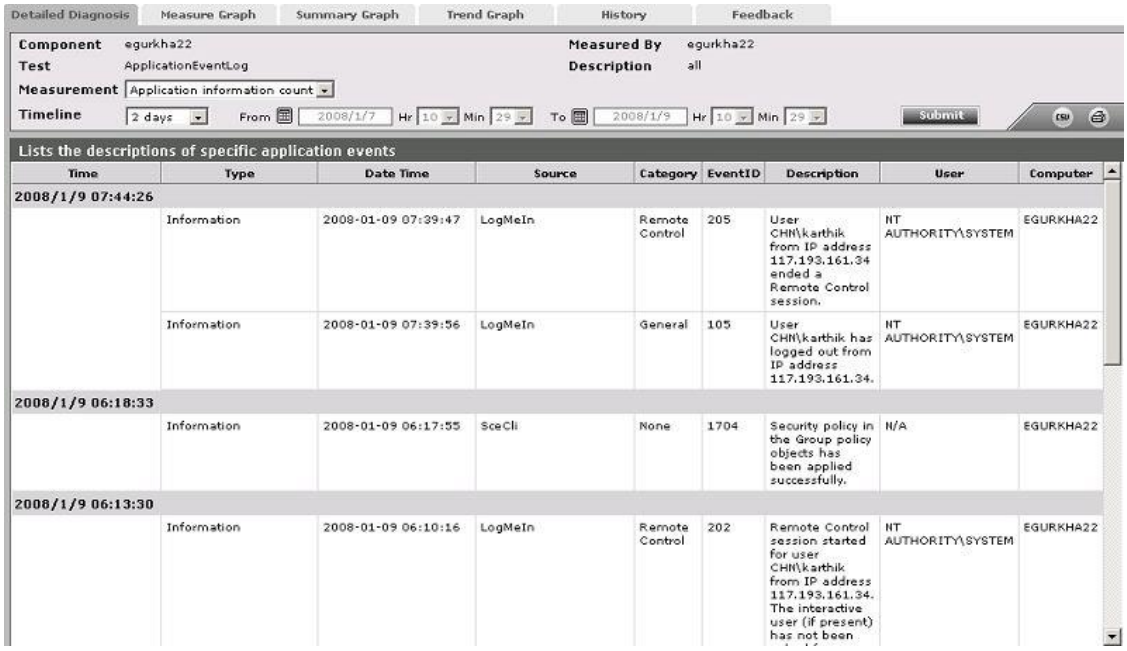

Figure 1.4: The detailed diagnosis of the Application information count measure

The filter policy for the ApplicationEventLog test, ApplicationEvents test, SystemEvents test, and SystemEventLog test typically comprises of a specific set of event sources, event IDs, and event descriptions to be monitored. This specification is expressed by the eG Enterprise system in the following format:

{Policyname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}

On the other hand, the filter policy for the SecurityLog test comprises of a specific set of event sources, event ids, and users to be monitored. This specification is expressed by the eG Enterprise system in the following format:

{Policyname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{users_to_be_included}:{users_to_be_excluded}

To add a new policy, do the following:

1. Click on the **Click here** hyperlink available just above the test configuration of the ApplicationEventLog test, ApplicationEvents test, SystemEvents test, SystemEventLog test, or SecurityLog test (see Figure 1.5).

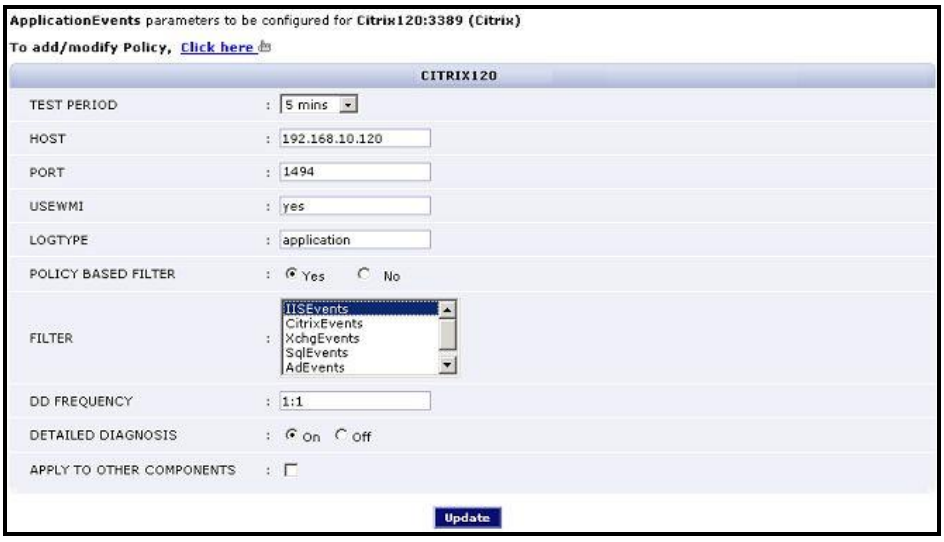

Figure 1.5: Configuring an ApplicationEvents test

2. Figure 1.6 will then appear listing the policies that pre-exist.

Figure 1.6: List of policies

3. To view the contents of a policy, click on the **View** button against the policy name. While a policy can be modified by clicking on the **Modify** button, it can be deleted using the **Delete** button. The default policy is **all**, which can only be viewed and **not modified** or **deleted**. The specification contained within this policy is: *all:none:all:none:all:none*.

4. To create a new policy, click on the **Add New Policy** button in Figure 1.6. Doing so invokes Figure 1.7, using which a new policy can be created.

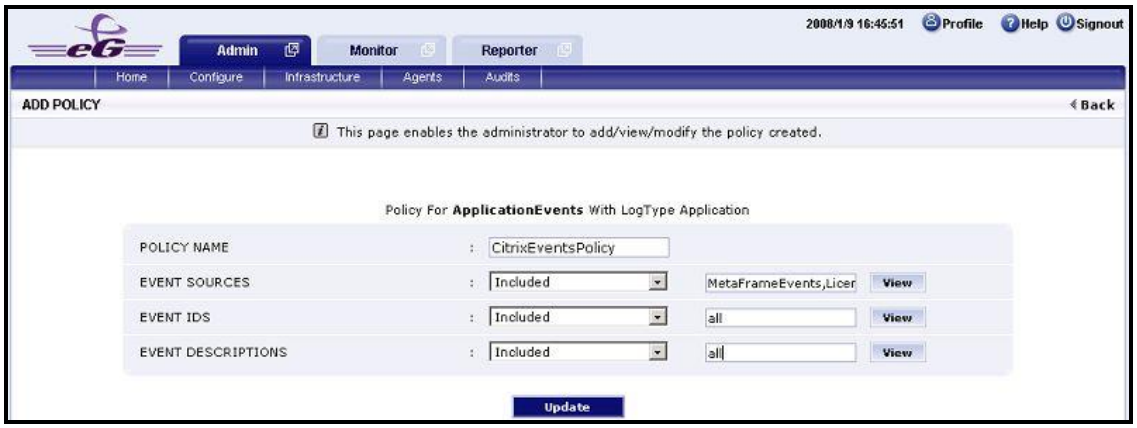

Figure 1.7: Adding a new filter policy

5. In Figure 1.7, first, provide a unique name against **POLICY NAME**.
6. To include one/more event sources for monitoring, select **Included** from the **EVENT SOURCES** drop-down list, and then specify a comma-separated list of event sources in the adjacent text box. If you require more space to specify the event sources, click on the **View** button next to the text box. This will invoke an **EVENT SOURCES INCLUDED** text area (see Figure 1.8), wherein the specification can be provided more clearly and comfortably.

Figure 1.8: Viewing the text area

7. To exclude specific event sources from monitoring, select **Excluded** from the **EVENT SOURCES** drop-down list, and then specify a comma-separated list of event sources to be excluded in the adjacent text box. If you require more space to specify the event sources, click on the **View** button next to the text box. This will invoke an **EVENT SOURCES EXCLUDED** text area, wherein the specification can be provided more clearly and comfortably.

Note:

At any given point in time, you can choose to either **Include** or **Exclude** event sources, but you cannot do both. If you have chosen to include event sources, then the eG Enterprise system automatically assumes that no event sources need be excluded. Accordingly, the *{event_sources_to_be_excluded}* section of the filter format mentioned above, will assume the value *none*. Similarly, if you have chosen to exclude specific event sources from monitoring, then the *{event_sources_to_be_included}* section of the format above will automatically take the value *all*, indicating that all event sources except the ones explicitly excluded, will be included for monitoring.

8. In the same way, select **Included** from the **EVENT IDS** list and then, provide a comma-separated list of event IDs to be monitored. For more space, click on the **View** button next to the text box, so that an **EVENT IDS INCLUDED** text area appears.
9. If you, on the other hand, want to exclude specific event IDs from monitoring, then first select **Excluded** from the **EVENT IDS** list box, and then provide a comma-separated list of event IDs to be excluded. For more space, click on the **View** button next to the text box, so that an **EVENT IDS EXCLUDED** text area appears.

Note:

At any given point in time, you can choose to either **Include** or **Exclude** event IDs, but you cannot do both. If you have chosen to include event IDs, then the eG Enterprise system automatically assumes that no event IDs need be excluded. Accordingly, the *{event_IDs_to_be_excluded}* section of the filter format mentioned above, will assume the value *none*. Similarly, if you have chosen to exclude specific event IDs from monitoring, then the *{event_IDs_to_be_included}* section of the format above will automatically take the value *all*, indicating that all event IDs except the ones explicitly excluded, will be included for monitoring.

10. Likewise, select **Included** from the **EVENT DESCRIPTIONS** list and then, provide a comma-separated list of event descriptions to be monitored. For more space, click on the **View** button next to the text box, so that an **EVENT DESCRIPTIONS INCLUDED** text area appears.
11. For excluding specific event descriptions from monitoring, first select **Excluded** from the **EVENT DESCRIPTIONS** list box, and then provide a comma-separated list of event descriptions to be excluded. For more space, click on the **View** button next to the text box, so that an **EVENT DESCRIPTIONS EXCLUDED** text area appears.

Note:

Instead of the complete event descriptions, wild card-embedded event description patterns can be provided as a comma-separated list in the **Included** or **Excluded** text boxes. For instance, to include all events that start with *st* and vi, your **Included** specification should be: *st*,vi**. Similarly, to exclude all events with descriptions ending with *ed* and *le*, your **Excluded** specification should be: **ed,*le*.

Note:

At any given point in time, you can choose to either **Include** or **Exclude** event descriptions/users, but you cannot do both. If you have chosen to include event descriptions/users, then the eG Enterprise system automatically assumes that no event descriptions/users need be excluded. Accordingly, the *{event_descriptions_to_be_excluded}* section or the *{users_to_be_excluded}* section (as the case may be) of the filter formats mentioned above, will assume the value *none*. Similarly, if you have chosen to exclude specific event descriptions/users from monitoring, then the *{event_descriptions_to_be_included}* section or the *{users_to_be_included}* section (as the case may be) of the formats above will automatically take the value *all*. This indicates that all event descriptions/users except the ones explicitly excluded, will be included for monitoring.

12. In case of the **SecurityLog** test however, you will not be required to include/exclude **EVENT DESCRIPTIONS**. Instead, an **EVENT USERS** field will appear, using which you need to configure users who need to be included/excluded from monitoring.
13. Finally, click the **Update** button.
14. The results of the configuration will then be displayed as depicted by Figure 1.9.

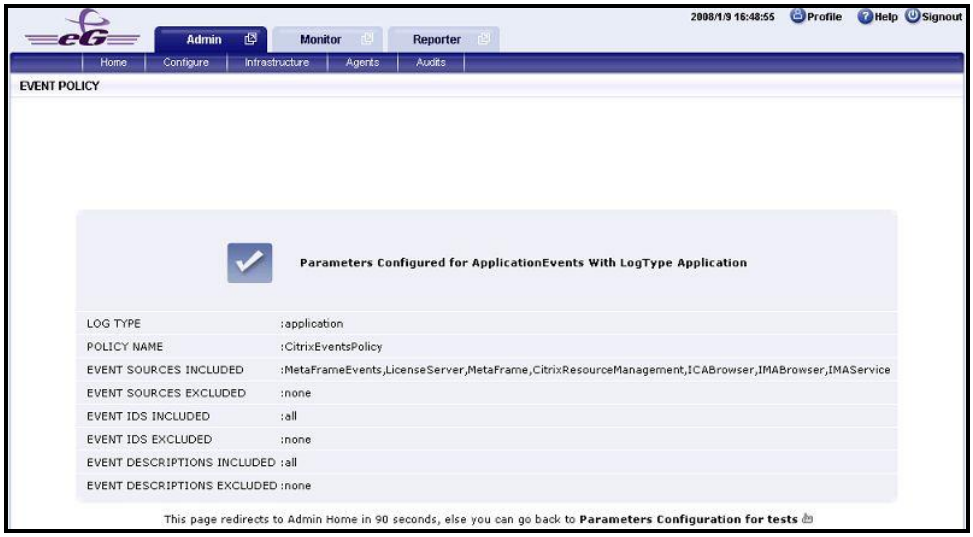

Figure 1.9: Results of the configuration

Note:

If you have configured a policy to **Include** a few/all events (sources/IDs/descriptions/users), and **Exclude** *none*, then, while reconfiguring that policy, you will find that the **Include** option is chosen by default from the corresponding drop-down list in Figure 1.7. On the other hand, if you have configured a policy to **Exclude** a few specific events and **Include** *all* events, then, while modifying that policy, you will find the **Exclude** option being the default selection in the corresponding drop-down list in Figure 1.7.

1.1.2 System Event Log Test

This test reports the statistical information about the system events generated by the target system.

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT –** Refers to the port used by the EventLog Service. Here it is null. 4. **LOGTYPE** – Refers to the type of event logs to be monitored. The default value is *application*. 5. **POLICY BASED FILTER** - Using this page, administrators can configure the event sources, event IDs, and event descriptions to be monitored by this test. In order to enable administrators to easily and accurately provide this specification, this page provides the following options: ➢ Manually specify the event sources, IDs, and descriptions in the **FILTER** text area, or, ➢ Select a specification from the predefined filter policies listed in the **FILTER** box For explicit, manual specification of the filter conditions, select the **NO** option against the **POLICY BASED FILTER** field. This is the default selection. To choose from the list of pre-configured filter policies, or to create a new filter policy and then associate the same with the test, select the **YES** option against the **POLICY BASED FILTER** field. 6. **FILTER** - If the **POLICY BASED FILTER** flag is set to **NO**, then a **FILTER** text area will appear, wherein you will have to specify the event sources, event IDs, and event descriptions to be monitored. This specification should be of the following format: *{Displayname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}*. For example, assume that the **FILTER** text area takes the value, *OS_events:all:Browse,Print:all:none:all:none*. Here: ➢ *OS_events* is the display name that will appear as a descriptor of the test in the monitor UI; ➢ *all* indicates that all the event sources need to be considered while monitoring. To monitor specific event sources, provide the source names as a comma-separated list. To ensure that none of the event sources are monitored, specify *none*. ➢ Next, to ensure that specific event sources are excluded from monitoring, provide a comma-separated list of source names. Accordingly, in our example, *Browse* and *Print* have been excluded from monitoring. Alternatively, you can use *all* to indicate that all the event sources have to be excluded from monitoring, or *none* to denote that none of the event sources need be excluded. ➢ In the same manner, you can provide a comma-separated list of event IDs that require monitoring. The *all* in our example represents that all the event IDs need to be considered while monitoring.

- Similarly, the *none* (following *all* in our example) is indicative of the fact that none of the event IDs need to be excluded from monitoring. On the other hand, if you want to instruct the eG Enterprise system to ignore a few event IDs during monitoring, then provide the IDs as a comma-separated list. Likewise, specifying *all* makes sure that all the event IDs are excluded from monitoring.
- The *all* which follows implies that all events, regardless of description, need to be included for monitoring. To exclude all events, use *none*. On the other hand, if you provide a comma-separated list of event descriptions, then the events with the specified descriptions will alone be monitored. Event descriptions can be of any of the following forms - *desc**, or *desc*, or **desc**,or *desc**, or *desc1***desc2*, etc. *desc* here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.
- In the same way, you can also provide a comma-separated list of event descriptions to be excluded from monitoring. Here again, the specification can be of any of the following forms: *desc**, or *desc*, or **desc**,or *desc**, or *desc1***desc2*, etc. *desc* here refers to any string that forms part of the description. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters. In our example however, none is specified, indicating that no event descriptions are to be excluded from monitoring. If you use *all* instead, it would mean that all event descriptions are to be excluded from monitoring.

By default, the **FILTER** parameter contains the value*: all:all:none:all:none:all:none*. Multiple filters are to be separated by semi-colons (;).

Note:

The event sources and event IDs specified here should be exactly the same as that which appears in the Event Viewer window.

On the other hand, if the **POLICY BASED FILTER** flag is set to **YES**, then a **FILTER** list box will appear, displaying the filter policies that pre-exist in the eG Enterprise system. A filter policy typically comprises of a specific set of event sources, event IDs, and event descriptions to be monitored. This specification is built into the policy in the following format:

{Policyname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{event_descriptions_to_be_included}:{event_descriptions_to_be_excluded}

To monitor a specific combination of event sources, event IDs, and event descriptions, you can choose the corresponding filter policy from the **FILTER** list box. Multiple filter policies can be so selected. Alternatively, you can modify any of the existing policies to suit your needs, or create a new filter policy. To facilitate this, a **Click here** link appears just above the test configuration section, once the **YES** option is chosen against **POLICY BASED FILTER**. Clicking on the **Click here** link leads you to a page where you can modify the existing policies or create a new one (refer to page 9). The changed policy or the new policy can then be associated with the test by selecting the policy name from the **FILTER** list box in this page.

	7. **USEWMI** - The eG agent can either use WMI to extract event log statistics or directly parse the event logs using event log APIs. If the **USEWMI** flag is **YES**, then WMI is used. If not, the event log APIs are used. This option is provided because on some Windows NT/2000 systems (especially ones with service pack 3 or lower), the use of WMI access to event logs can cause the CPU usage of the WinMgmt process to shoot up. On such systems, set the **USEWMI** parameter value to **NO**. **On the other hand, when monitoring systems that are operating on any other flavor of Windows (say, Windows 2003/XP/2008/7/Vista/12), the USEWMI flag should always be set to 'Yes'.** 8. **STATELESS ALERTS** - Typically, the eG manager generates email alerts only when the state of a specific measurement changes. A state change typically occurs only when the threshold of a measure is violated a configured number of times within a specified time window. While this ensured that the eG manager raised alarms only when the problem was severe enough, in some cases, it may cause one/more problems to go unnoticed, just because they did not result in a state change. For example, take the case of the EventLog test. When this test captures an error event for the very first time, the eG manager will send out a **CRITICAL** email alert with the details of the error event to configured recipients. Now, the next time the test runs, if a different error event is captured, the eG manager will keep the state of the measure as **CRITICAL**, but will not send out the details of this error event to the user; thus, the second issue will remain hidden from the user. To make sure that administrators do not miss/overlook critical issues, the eG Enterprise monitoring solution provides the **stateless alerting** capability. To enable this capability for this test, set the **STATELESS ALERTS** flag to **Yes**. This will ensure that email alerts are generated for this test, regardless of whether or not the state of the measures reported by this test changes. 9. **EVENTS DURING RESTART** - By default, the **EVENTS DURING RESTART** flag is set to **Yes**. This ensures that whenever the agent is stopped and later started, the events that might have occurred during the period of non-availability of the agent are included in the number of events reported by the agent. Setting the flag to **No** ensures that the agent, when restarted, ignores the events that occurred during the time it was not available. 10. **DDFORINFORMATION** – eG Enterprise also provides you with options to restrict the amount of storage required for event log tests. Towards this end, the **DDFORINFORMATION** and **DDFORWARNING** flags have been made available in this page. By default, both these flags are set to **Yes**, indicating that by default, the test generates detailed diagnostic measures for information events and warning events. If you do not want the test to generate and store detailed measures for information events, set the **DDFORINFORMATION** flag to **No**. 11. **DDFORWARNING** – To ensure that the test does not generate and store detailed measures for warning events, set the **DDFORWARNING** flag to **No**.

<table>
<tr><td></td><td colspan="3">12. DDFREQUENCY - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is 1:1. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying none against DDFREQ.
13. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option. To disable the capability, click on the Off option.
The option to selectively enabled/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
➢ The eG manager license should allow the detailed diagnosis capability
➢ Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the FILTER configured</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>System Errors:
This refers to the number of system error events generated during the last execution of the test.</td><td>Number</td><td>A very low value (zero) indicates that the system is in healthy state and all Windows services and low level drivers are running without any potential problems.
An increasing trend or a high value indicates the existence of problems such as loss of functionality or data in one or more Windows services and low level drivers.
Please check the System Logs in the Event Log Viewer for more details.</td></tr>
<tr><td>System information messages:
This refers to the number of service-related and driver-related information events that were generated during the test's last execution.</td><td>Number</td><td>A change in value of this measure may indicate infrequent but successful operations performed by one or more applications.
Please check the System Logs in the Event Log Viewer for more details.</td></tr>
</table>

	System warnings: This refers to the number of service-related and driver-related warnings generated in the during the test's last execution.	Number	A high value of this measure indicates problems that may not have an immediate impact, but may cause future problems in one or more Windows servers and low level drivers. Please check the System Logs in the Event Log Viewer for more details.
	System critical errors: Indicates the number of critical events that were generated when the test was last executed.	Number	A critical event is one that a system cannot automatically recover from. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. A very low value (zero) indicates that the system is in a healthy state and is running smoothly without any potential problems. An increasing trend or high value indicates the existence of fatal/irrepairable problems in the system. The detailed diagnosis of this measure describes all the critical system events that were generated during the last measurement period. Please check the System Logs in the Event Log Viewer for more details.
	System verbose: Indicates the number of verbose events that were generated when the test was last executed.	Number	Verbose logging provides more details in the log entry, which will enable you to troubleshoot issues better. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. The detailed diagnosis of this measure describes all the verbose events that were generated during the last measurement period. Please check the System Logs in the Event Log Viewer for more details.

The detailed diagnosis of the *System errors* measure, provides detailed descriptions of the system errors that occurred on the host during the last measurement period.

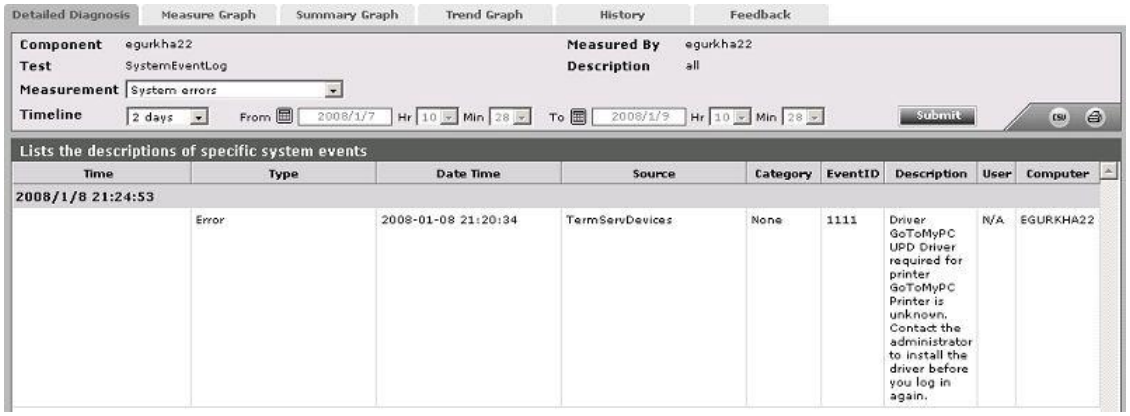

Figure 1.10: The detailed diagnosis of the System errors measure

The detailed diagnosis of the *System information messages* measure, provides detailed descriptions of the information events that occurred on the host during the last measurement period.

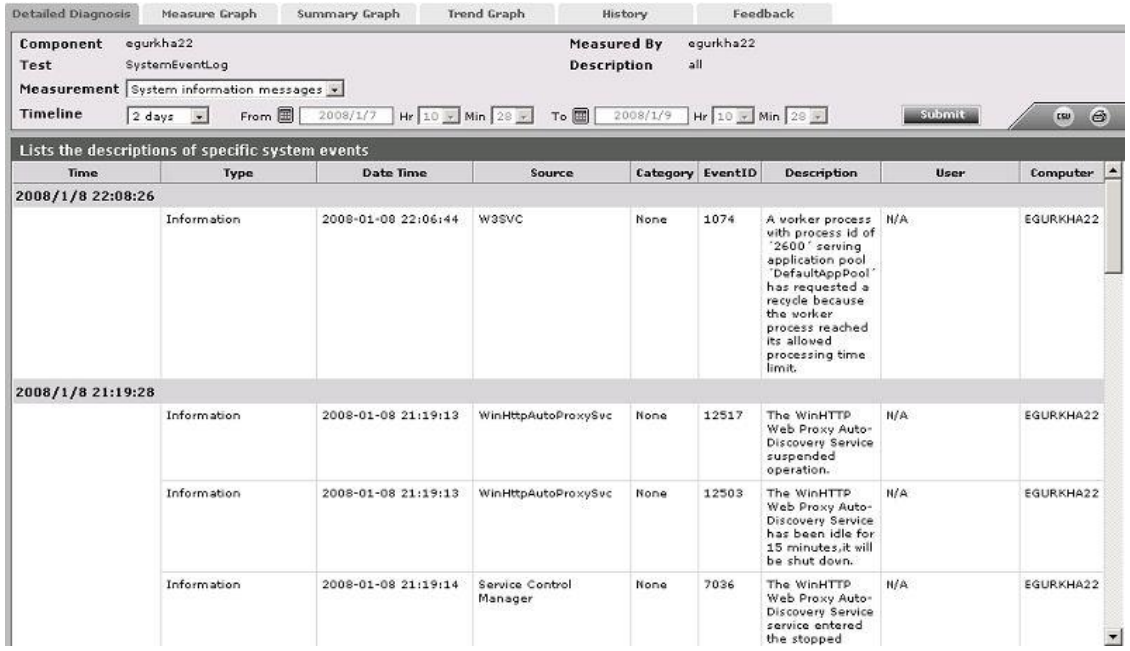

Figure 1.11: The detailed diagnosis of the System information messages measure

The detailed diagnosis of the *System warnings* measure, provides detailed descriptions of the warning events that occurred on the host during the last measurement period.

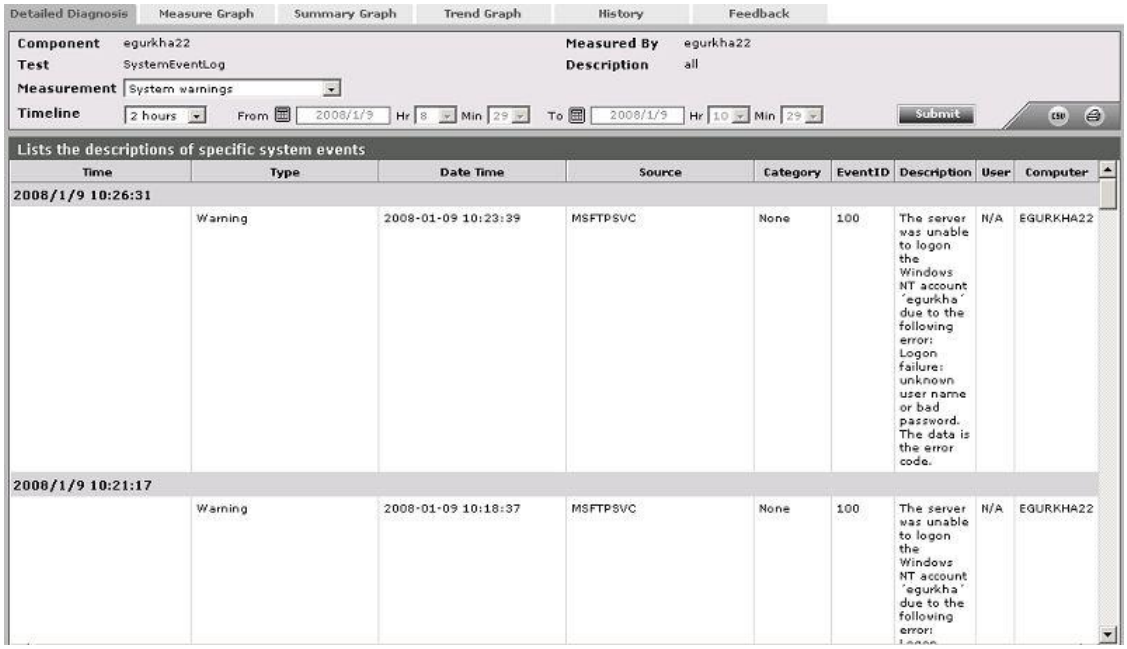

Figure 1.12: The detailed diagnosis of the System warnings measure

1.1.3 Security Log Test

The Security Log test reports statistics relating to the Windows security log audits. **Note that this test will not work on Windows Vista.**

Purpose	Reports statistics relating the Windows security log audits
Target of the test	Any Windows host system
Agent deploying the test	An internal agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT –** Refers to the port used by the EventLog Service. Here it is null. 4. **LOGTYPE** - Refers to the type of event logs to be monitored. The default value is *security*. 5. **SUCCESSEVENTSINDD** - By default, this parameter displays *none*, indicating that by default none of the successful log audits will be reflected in the detailed diagnosis. If you set this parameter to, say 10, then the test will display only the 10 most recent successful log audits in the detailed diagnosis page. Setting this parameter to *all*, on the other hand will make sure that all successful log audits are listed in the detailed diagnosis. 6. **FAILUREEVENTSINDD** - By default, this parameter displays *all*, indicating that by default all the failed log audits will be reflected in the detailed diagnosis. If you set this parameter to, say 10, then the test will display only the 10 most recent log audits that failed, in the detailed diagnosis page. Setting this parameter to *none*, on the other hand will make sure that none of the failed log audits are listed in the detailed diagnosis. 7. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against **DD FREQUENCY**. 8. **USEWMI** - The eG agent can either use WMI to extract event log statistics or directly parse the event logs using event log APIs. If the **USEWMI** flag is **YES**, then WMI is used. If not, the event log APIs are used. This option is provided because on some Windows NT/2000 systems (especially ones with service pack 3 or lower), the use of WMI access to event logs can cause the CPU usage of the WinMgmt process to shoot up. On such systems, set the **USEWMI** parameter value to **NO**. **On the other hand, when monitoring systems that are operating on any other flavor of Windows (say, Windows 2003/XP/2008/7/Vista/12), the USEWMI flag should always be set to 'Yes'.** 9. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enabled/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: 10. The eG manager license should allow the detailed diagnosis capability 11. Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

12. **POLICY BASED FILTER** - Using this page, administrators can configure the event sources, event IDs, and event descriptions to be monitored by this test. In order to enable administrators to easily and accurately provide this specification, this page provides the following options:

    - Manually specify the event sources, IDs, and users in the **FILTER** text area, or,
    - Select a specification from the predefined filter policies listed in the **FILTER** box

    For explicit, manual specification of the filter conditions, select the **NO** option against the **POLICY BASED FILTER** field. To choose from the list of pre-configured filter policies, or to create a new filter policy and then associate the same with the test, select the **YES** option against the **POLICY BASED FILTER** field. This is the default selection.

13. **FILTER** - If the **POLICY BASED FILTER** flag is set to **NO**, then a **FILTER** text area will appear, wherein you will have to specify the event sources, event IDs, and event users to be monitored. This specification should be of the following format:
    {Displayname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{users_to_be_included}:{ users_to_be_excluded}. For example, assume that the **FILTER** text area takes the value, *OS_events:all:Browse,Print:all:none:all:none*. Here:

    - *OS_events* is the display name that will appear as a descriptor of the test in the monitor UI;
    - *all* indicates that all the event sources need to be considered while monitoring. To monitor specific event sources, provide the source names as a comma-separated list. To ensure that none of the event sources are monitored, specify *none*.
    - Next, to ensure that specific event sources are excluded from monitoring, provide a comma-separated list of source names. Accordingly, in our example, *Browse* and *Print* have been excluded from monitoring. Alternatively, you can use *all* to indicate that all the event sources have to be excluded from monitoring, or *none* to denote that none of the event sources need be excluded.
    - In the same manner, you can provide a comma-separated list of event IDs that require monitoring. The *all* in our example represents that all the event IDs need to be considered while monitoring.
    - Similarly, the *none* (following *all* in our example) is indicative of the fact that none of the event IDs need to be excluded from monitoring. On the other hand, if you want to instruct the eG Enterprise system to ignore a few event IDs during monitoring, then provide the IDs as a comma-separated list. Likewise, specifying *all* makes sure that all the event IDs are excluded from monitoring.

<table>
<tr><td></td><td colspan="3">

In the same way, you can also ensure that events generated by specific users on the target host are alone tracked by providing a comma-separated list of users to be monitored – for example, john,elvis. In our example however, all is specified, indicating that all users need be monitored.
You can similarly indicate if specific users need to be excluded from monitoring. In our example however, none is provided to ensure that no users are excluded from monitoring.

By default, the FILTER parameter contains the value: all:all:none:all:none:all:none. Multiple filters are to be separated by semi-colons (;).
Note:
The event sources and event IDs specified here should be exactly the same as that which appears in the Event Viewer window.
On the other hand, if the POLICY BASED FILTER flag is set to YES, then a FILTER list box will appear, displaying the filter policies that pre-exist in the eG Enterprise system. A filter policy typically comprises of a specific set of event sources, event IDs, and users to be monitored. This specification is built into the policy in the following format:
{Policyname}:{event_sources_to_be_included}:{event_sources_to_be_excluded}:{event_IDs_to_be_included}:{event_IDs_to_be_excluded}:{users_to_be_included}:{users_to_be_excluded}
To monitor a specific combination of event sources, event IDs, and users, you can choose the corresponding filter policy from the FILTER list box. Multiple filter policies can be so selected. Alternatively, you can modify any of the existing policies to suit your needs, or create a new filter policy. To facilitate this, a Click here link appears just above the test configuration section, once the YES option is chosen against POLICY BASED FILTER. Clicking on the Click here link leads you to a page where you can modify the existing policies or create a new one (refer to page 9). The changed policy or the new policy can then be associated with the test by selecting the policy name from the FILTER list box in this page.
</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the server being monitored</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Successful audits:
Indicates the number of successful audits of windows security logs.</td><td>Number</td><td>The detailed diagnosis of this measure, if enabled, provides the details of the successful log audits.</td></tr>
<tr><td>Failure audits:
Indicates the number of windows security log audits that failed.</td><td>Number</td><td>The detailed diagnosis of this measure, if enabled, provides the details of the failed log audits.</td></tr>
</table>

The detailed diagnosis of the *Successful audits* measure, if enabled, provides the details of the successful log audits.

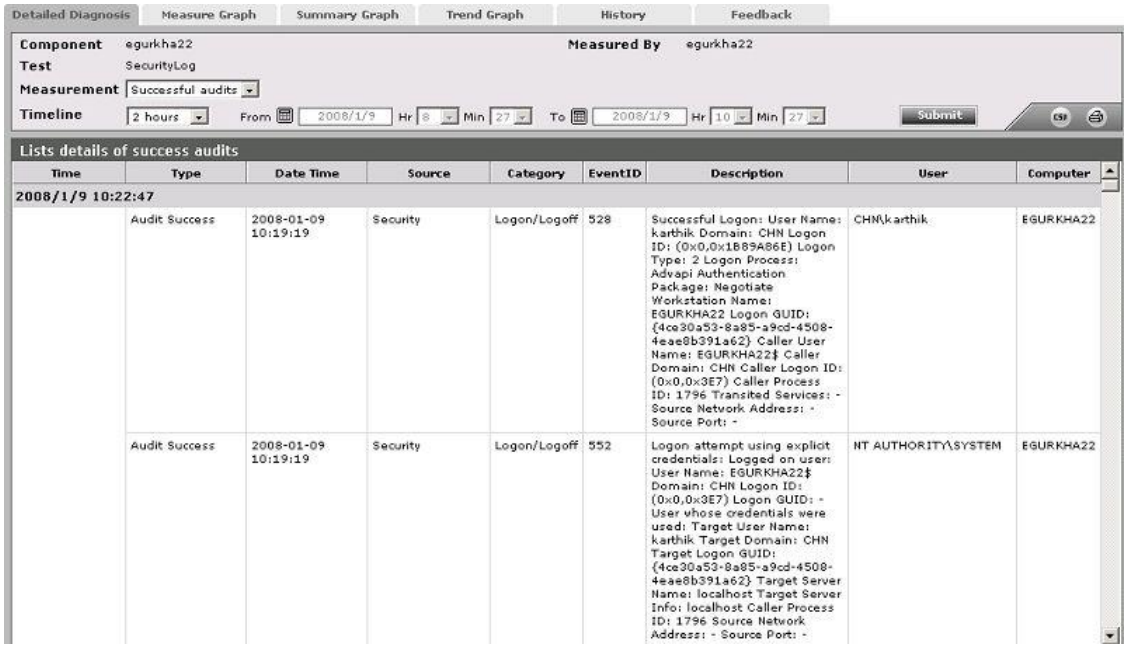

Figure 1.13: The detailed diagnosis of the Successful audits measure

1.1.4 Event Log Test

This test reports the statistical information about the events generated by various applications and windows services and drivers in the target system. This test is disabled by default. To enable the test, go to the **ENABLE / DISABLE TESTS** page using the menu sequence : Agents -> Tests -> Enable/Disable, pick *Event Log* as the **Component type**, *Performance* as the **Test type**, choose the test from the **DISABLED TESTS** list, and click on the **>>** button to move the test to the **ENABLED TESTS** list. Finally, click the **Update** button.

Purpose	Reports information about the events generated by various applications and windows services and drivers.
Target of the test	Any host system
Agent deploying the test	An internal agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT –** Refers to the port used by the EventLog Service. Here it is null. 4. **EVENTHOST** – Is the same as the **HOST** 5. **EVENTSRC** - Enter the specific events to be monitored in the **EVENTSRC** text box. The name of the event source can be obtained from the Event Viewer window that appears on following the menu sequence: Start -> Programs -> Administrative Tools -> Event Viewer (If the Programs menu does not contain the Administrative Tools option, then check Start->Settings ->Control Panel for the same). The value that appears in the Source column of this window should be used to specify the **EVENTSRC** parameter. By default, "All" will be displayed against **EVENTSRC** indicating that all events will be monitored by default. While specifying multiple events, make sure that they are separated by commas (,). 6. **EXCLUDEDSRC** - If specific events are to be excluded from monitoring, then specify the events to be excluded in the **EXCLUDEDSRC** text box, as a comma-separated list. 7. **USEWMI** - The eG agent can either use WMI to extract event log statistics or directly parse the event logs using event log APIs. If the **USEWMI** flag is **YES**, then WMI is used. If not, the event log APIs are used. This option is provided because on some Windows NT/2000 systems (especially ones with service pack 3 or lower), the use of WMI access to event logs can cause the CPU usage of the WinMgmt process to shoot up. On such systems, set the **USEWMI** parameter value to **NO**. **On the other hand, when monitoring systems that are operating on any other flavor of Windows (say, Windows 2003/XP/2008/7/Vista/12), the USEWMI flag should always be set to 'Yes'.** 8. **STATELESS ALERTS** - Typically, the eG manager generates email alerts only when the state of a specific measurement changes. A state change typically occurs only when the threshold of a measure is violated a configured number of times within a specified time window. While this ensured that the eG manager raised alarms only when the problem was severe enough, in some cases, it may cause one/more problems to go unnoticed, just because they did not result in a state change. For example, take the case of the EventLog test. When this test captures an error event for the very first time, the eG manager will send out a **CRITICAL** email alert with the details of the error event to configured recipients. Now, the next time the test runs, if a different error event is captured, the eG manager will keep the state of the measure as **CRITICAL**, but will not send out the details of this error event to the user; thus, the second issue will remain hidden from the user. To make sure that administrators do not miss/overlook critical issues, the eG Enterprise monitoring solution provides the **stateless alerting** capability. To enable this capability for this test, set the **STATELESS ALERTS** flag to **Yes**. This will ensure that email alerts are generated for this test, regardless of whether or not the state of the measures reported by this test changes.

<table>
<tr><td></td><td colspan="3">9. DD FREQUENCY - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is 1:1. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying none against DDFREQ.
10. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option. To disable the capability, click on the Off option.
The option to selectively enabled/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
➢ The eG manager license should allow the detailed diagnosis capability
➢ Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for server being monitored</td></tr>
<tr><td rowspan="4">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Application errors:
This refers to the number of application error events that were generated.</td><td>Number</td><td>A very low value (zero) indicates that the system is in a healthy state and all applications are running smoothly without any potential problems.
An increasing trend or high value indicates the existence of problems like loss of functionality or data in one or more applications.
Please check the Application Logs in the Event Log Viewer for more details.</td></tr>
<tr><td>Appplication information messages:
This refers to the number of application information events generated when the test was last executed.</td><td>Number</td><td>A change in the value of this measure may indicate infrequent but successful operations performed by one or more applications.
Please check the Application Logs in the Event Log Viewer for more details.</td></tr>
<tr><td>Application warnings:
This refers to the number of warnings that were generated when the test was last executed.</td><td>Number</td><td>A high value of this measure indicates application problems that may not have an immediate impact, but may cause future problems in one or more applications.
Please check the Application Logs in the Event Log Viewer for more details.</td></tr>
</table>

	Application critical errors: Indicates the number of critical events that were generated when the test was last executed.	Number	A critical event is one that an application or a component cannot automatically recover from. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. A very low value (zero) indicates that the system is in a healthy state and all applications are running smoothly without any potential problems. An increasing trend or high value indicates the existence of fatal/irrepairable problems in one or more applications. The detailed diagnosis of this measure describes all the critical application events that were generated during the last measurement period. Please check the Application Logs in the Event Log Viewer for more details.
	Application verbose: Indicates the number of verbose events that were generated when the test was last executed.	Number	Verbose logging provides more details in the log entry, which will enable you to troubleshoot issues better. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. The detailed diagnosis of this measure describes all the verbose events that were generated during the last measurement period. Please check the Application Logs in the Event Log Viewer for more details.
	System errors: This refers to the number of system error events generated during the last execution of the test.	Number	A very low value (zero) indicates that the system is in healthy state and all Windows services and low level drivers are running without any potential problems. An increasing trend or a high value indicates the existence of problems such as loss of functionality or data in one or more Windows services and low level drivers. Please check the System Logs in the Event Log Viewer for more details.

	System information messages: This refers to the number of service-related and driver-related information events that were generated during the test's last execution.	Number	A change in value of this measure may indicate infrequent but successful operations performed by one or more applications. Please check the System Logs in the Event Log Viewer for more details.
	System warnings: This refers to the number of service-related and driver-related warnings generated in the during the test's last execution.	Number	A high value of this measure indicates problems that may not have an immediate impact, but may cause future problems in one or more Windows servers and low level drivers. Please check the System Logs in the Event Log Viewer for more details.
	System critical errors: Indicates the number of critical events that were generated when the test was last executed.	Number	A critical event is one that a system cannot automatically recover from. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. A very low value (zero) indicates that the system is in a healthy state and is running smoothly without any potential problems. An increasing trend or high value indicates the existence of fatal/irrepairable problems in the system. The detailed diagnosis of this measure describes all the critical system events that were generated during the last measurement period. Please check the System Logs in the Event Log Viewer for more details.

	System verbose: Indicates the number of verbose events that were generated when the test was last executed.	Number	Verbose logging provides more details in the log entry, which will enable you to troubleshoot issues better. This measure is applicable only for Windows 2008/Windows Vista/Windows 7 systems. The detailed diagnosis of this measure describes all the verbose events that were generated during the last measurement period. Please check the System Logs in the Event Log Viewer for more details.

Note:

The **EVENTSRC** specified should be exactly the same as that which appears in the Event Viewer window.

Note:

The **STATELESS ALERTING** capability is currently available for the following tests alone, by default:

- EventLog test
- ApplicationEventLog test
- SystemEventLog test
- ApplicationEvents test
- SystemEvents test
- SecurityLog test
- Account Management Events test

If need be, you can enable the **stateless alerting** capability for other tests. To achieve this, follow the steps given below:

- Login to the eG manager host.
- Edit the **eg_specs.ini** file in the **<EG_INSTALL_DIR>\manager\config** directory.
- Locate the test for which the **Stateless Alarms** flag has to be enabled.
- Insert the entry, **-statelessAlerts yes**, into the test specification as depicted below:

```
EventLogTest::$hostName:$portNo=$hostName, -auto, -host $hostName -port
$portNo -eventhost $hostIp -eventsrc all -excludedSrc none -useWmi yes -
statelessAlerts yes -ddFreq 1:1 -rptName $hostName, 300
```

- Finally, save the file.
- If need be, you can change the status of the **statelessAlerts** flag by reconfiguring the test in the eG administrative interface.

Once the **stateless alerting capability** is enabled for a test (as discussed above), you will find that everytime the test reports a problem, the eG manager does the following:

- Closes the alarm that pre-exists for that problem;
- Sends out a normal alert indicating the closure of the old problem;
- Opens a new alarm and assigns a new alarm ID to it;
- Sends out a fresh email alert to the configured users, intimating them of the new issue.

In a redundant manager setup, the secondary manager automatically downloads the updated **eg_specs.ini** file from the primary manager, and determines whether the stateless alerting capability has been enabled for any of the tests reporting metrics to it. If so, everytime a threshold violation is detected by such a test, the secondary manager will perform the tasks discussed above for the problem reported by that test. Similarly, the primary manager will check whether the stateless alert flag has been switched on for any of the tests reporting to it, and if so, will automatically perform the above-mentioned tasks whenever those tests report a deviation from the norm.

Note:

- Since alerts will be closed after every measurement period, alarm escalation will no longer be relevant for tests that have **statelessAlerts** set to **yes**.
- For tests with **statelessAlerts** set to **yes**, **statelessAlerts** will apply for all measurements of that test (i.e., it will not be possible to only have one of the measurements with stateless alerts and others without).
- If **statelessAlerts** is set to **yes** for a test, an alarm will be opened during one measurement period (if a threshold violation happens) and will be closed prior to the next measurement period. This way, if a threshold violation happens in successive measurement periods, there will be one alarm per measurement period. This will reflect in all the corresponding places in the eG Enterprise system. For example, multiple alerts in successive measurement periods will result in multiple trouble tickets being opened (one for each measurement period). Likewise, the alarm history will also show alarms being opened during a measurement period and closed during the next measurement period.

Chapter 2

Conclusion

This document has described in detail the monitoring paradigm used and the measurement capabilities of the eG Enterprise suite of products with respect to **Event Logs**. For details of how to administer and use the eG Enterprise suite of products, refer to the user manuals.

We will be adding new measurement capabilities into the future versions of the eG Enterprise suite. If you can identify new capabilities that you would like us to incorporate in the eG Enterprise suite of products, please contact support@eginnovations.com. We look forward to your support and cooperation. Any feedback regarding this manual or any other aspects of the eG Enterprise suite can be forwarded to feedback@eginnovations.com.